FIPS 140-2 Security Policy

BlackBerry® Cryptographic Kernel
Versions 3.8.0.18, 3.8.0.20, 3.8.0.23, and 3.8.0.24

Document Version 1.8

BlackBerry Security Team
Research In Motion®

Document and Contact Information

Version	Date	Author	Description
1.0	07 July 2004	David MacFarlane	Document creation.
1.1	27 August 2004	David MacFarlane	Moved to new document template.
1.2	8 September 2004	David MacFarlane	Updated module version number.
1.3	9 September 2004	David MacFarlane	Updated per conformance testing feedback and included version 3.8.0.20 of the module.
1.4	4 October 2004	David MacFarlane	Updated algorithm certificate numbers and included version 3.8.0.23 of the module.
1.5	20 December 2004	David MacFarlane	Updated per CMVP review comments.
1.6	22 December 2004	David MacFarlane	Updated per CMVP review comments.
1.7	05 January 2005	David MacFarlane	Updated per CMVP review comments.
1.8	10 January 2005	David MacFarlane	Included module version 3.8.0.24.

Contact	Corporate Office
BlackBerry Security Team BlackBerrySecurity@rim.com (519) 888-7465 ext. 2921	**Research In Motion Limited** 175 Columbia Street West Waterloo ON Canada N2L 5Z5 www.rim.com

Contents

List of Tables

List of Figures

Introduction

BlackBerry is the leading wireless enterprise solution that allows users to stay connected with secure, wireless access to e-mail, corporate data, phone, web, and organiser features. BlackBerry is a totally integrated package that includes hardware, software, and service, providing a complete end-to-end solution. More information on the BlackBerry wireless solution is available at http://www.blackberry.com/.

Each BlackBerry Wireless Handheld™ contains the BlackBerry Cryptographic Kernel[1], a software cryptographic module that provides the cryptographic functionality required for basic operation of the handheld. For the purposes of FIPS 140-2 conformance testing the BlackBerry Cryptographic Kernel was executed on the BlackBerry 7230 Wireless Handheld™ and, per FIPS 140-2 Implementation Guidance G.5, remains FIPS-compliant when executed on other BlackBerry handhelds. The BlackBerry Cryptographic Kernel is hereafter referred to as *cryptographic module* or *module*.

Introduced in version 3.8.0.0 and present in all subsequent versions of the module is additional cryptographic functionality required for new features introduced in BlackBerry handheld software version 4.0, namely enhanced protection for wireless communications between the handheld and the BlackBerry Enterprise Server™, protection of data stored on the handheld, and wireless enterprise activation and provisioning. Wireless communications between the handheld and the BlackBerry Enterprise Server™ are protected with Triple DES encryption and BlackBerry handheld software version 4.0 introduces support for alternatively protecting these communications with AES-256 encryption[2]. Similarly, BlackBerry handheld software version 4.0 introduces support for the use of AES-256 encryption to protect data stored on the handheld. Also introduced is key agreement functionality to support wireless activation and provisioning of the handheld[2].

[1] Excludes RIM 850™, RIM 950™, RIM 857™, and RIM 957™ wireless handhelds.
[2] Requires BlackBerry handheld and BlackBerry Enterprise Server™ software version 4.0 or higher.

Cryptographic Module Specification

The cryptographic module is a software module that implements the following FIPS-Approved security functions [3]:

- **AES-256** (encrypt and decrypt), as specified in FIPS 197. The implementation supports the CBC mode of operation and has been awarded AES validation certificate #177, http://csrc.nist.gov/cryptval/aes/aesval.html.
- **Triple DES** (encrypt and decrypt), as specified in FIPS 46-3. The implementation supports the CBC mode of operation and has been awarded Triple DES validation certificate #281, http://csrc.nist.gov/cryptval/des/tripledesval.html.
- **RSA PKCS#1** (signature verification), as specified in PKCS #1, version 2.1. The implementation has been awarded RSA validation certificate #22, http://csrc.nist.gov/cryptval/dss/rsaval.html.
- **SHA-1**, **SHA-256**, and **SHA-512**, as specified in FIPS 180-2. The implementation has been awarded SHS validation certificate #264, http://csrc.nist.gov/cryptval/shs/shaval.htm.
- **HMAC SHA-1**, **HMAC SHA-256**, and **HMAC SHA-512**, as specified in FIPS 198. The implementation has been awarded HMAC validation certificate #1, http://csrc.nist.gov/cryptval/mac/hmacval.htm.
- **FIPS 186-2 RNG**, as specified in FIPS 186-2 Appendix 3.1. The implementation uses SHA-1 as the function *G* and has been awarded RNG validation certificate #27, http://csrc.nist.gov/cryptval/rng/rngval.html.

The module implements the following non-Approved security functions that, per *FIPS 140-2 Annex D: Approved Key Establishment Techniques for FIPS PUB 140-2*, may presently be used in a FIPS-Approved mode of operation:

- **EC Diffie-Hellman** (key agreement), as specified in IEEE P1363 Draft 13.
- **EC MQV**, as specified in IEEE P1363 Draft 13.

The module does not have a non-Approved mode of operation and, consequently, always operates in a FIPS-Approved mode of operation.

The physical boundary of the module is the physical boundary of the handheld that executes the module and is shown in the following figure:

[3] A security function is FIPS-Approved if it is explicitly listed in *FIPS 140-2 Annex A: Approved Security Functions for FIPS PUB 140-2*.

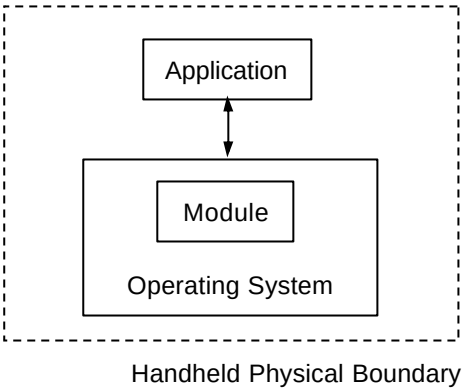

Figure 1. Physical Boundary

Determining the Module Version

The operator may determine the version of the module on a handheld by performing the following operations:

1. Click the **Options** icon in the Home screen. The **Options** list appears.
2. From the **Options** list, click the **About** item. The About screen appears and displays the module version, e.g. “Cryptographic Kernel v3.8.0.18”.

Cryptographic Module Ports and Interfaces

The module ports correspond to the physical ports on the BlackBerry handheld executing the module software, and the module interfaces correspond to the logical interfaces to the module. The following table describes the ports and interfaces implemented by the BlackBerry 7230™.

Table 1. Implementation of FIPS 140-2 Interfaces

FIPS 140-2 Interface	Module Ports	Module Interfaces
Data Input	Keyboard, microphone, USB port, headset jack, wireless modem	Input parameters of module function calls
Data Output	Speaker, USB port, headset jack, wireless modem	Output parameters of module function
Control Input	Keyboard, USB port, trackwheel, escape button, backlight button, phone button	Module function calls
Status Output	USB port, LCD screen, LED	Return codes of module function calls
Power Input	USB port	Not applicable
Maintenance	Not supported	Not supported

Roles, Services, and Authentication

The module does not support multiple or concurrent operators and is intended for use by a single operator, thus it always operates in a single-user mode of operation. Operators are not required to authenticate to the module.

The module supports a User and Crypto Officer role. The module does not support a Maintenance role. Role selection is performed implicitly and is dependent on the service performed by the operator. The following table describes the services that are available to the operator:

- **Reset** – Resets the module. The module may be reset by pressing the **Alt** key + **Right Shift** key + **Backspace** key combination or power cycling the module.
- **View Status** – Displays the status of the module.
- **Inject Master Key** – Replaces the existing Master Key with a new Master Key. The new Master Key is created outside of the cryptographic boundary for this service.
- **Perform Key Agreement** – Creates a new Master Key and uses it to replace the existing Master Key. The new Master Key is created by performing key agreement with the BlackBerry Enterprise Server.
- **Inject PIN Master Key** – Replaces the existing PIN Master Key with a new PIN Master Key. The new PIN Master Key is created outside the cryptographic boundary and is encrypted for input into the module for this service.
- **Generate Session Key** – Generates a Session Key or a PIN Session Key. This service is performed automatically on behalf of the operator during the **Encrypt Data** service.
- **Encrypt Data** – Encrypts data that is to be sent from the handheld. A Session Key is automatically generated via the **Generate Session Key** service and used to encrypt the data. The Session Key is encrypted with the Master Key and then the encrypted data and encrypted Session Key are ready for transmission.
- **Decrypt Data** – Decrypts data that has been received by the handheld. The encrypted Session Key is decrypted with the Master Key and is then used to decrypt the data. This service is performed automatically on behalf of the operator.
- **Generate HMAC** – Generates a message authentication code.
- **Perform Self-Tests** – Executes the module self-tests, as described in Self-Tests on page 9.

The following table summarises implicit role selection based on services and the associated access to critical security parameters (CSPs):

Table 2. Role Selection and CSP Access by Service

Service	Role Implicitly Selected	Affected Keys and CSPs	Access to Keys and CSPs
Reset	User	N/A	N/A
Power On	User	N/A	N/A
Power Off	User	N/A	N/A

Service	Role Implicitly Selected	Affected Keys and CSPs	Access to Keys and CSPs
View Status	User	N/A	N/A
Inject Master Key	Crypto Officer	Master Key	Write
Perform Key Agreement	Crypto Officer	Master Key	Write
		EC MQV Key Pair	Write, Execute
		EC DH Key Pair	Write, Execute
Inject PIN Master Key	Crypto Officer	PIN Master Key	Write
Generate Session Key	User	Session Key	Write
		PIN Session Key	Write
Encrypt Data	User	Master Key / PIN Master Key	Execute
		Session Key / PIN Session Key	Write, Execute
Decrypt Data	User	Master Key / PIN Master Key	Execute
		Session Key / PIN Session Key	Execute
Generate HMAC	User	HMAC SHA-1 Key / HMAC SHA-256 Key / HMAC SHA-512 Key	Execute
Perform Self-Tests	User	N/A	N/A

Physical Security

The BlackBerry handheld that executes the module meets the FIPS 140-2 Level 1 physical security requirements.

Cryptographic Keys and Critical Security Parameters

The following table describes the cryptographic keys, key components, and CSPs utilised by the module.

Table 3. Cryptographic Keys and CSPs

Key / CSP	Description
Master Key	A Triple DES or AES-256 key used to encrypt and decrypt Session Keys. The Master Key can be generated outside of the cryptographic boundary and input into the module, or created cooperatively with the BlackBerry Enterprise Server™ through EC Diffie-Hellman or EC MQV key agreement.
Session Key	A Triple DES or AES-256 key used to encrypt and decrypt data. The module generates Session Keys using the implemented FIPS 186-2 RNG.
PIN Master Key	A Master Key that is specifically a Triple DES key used to encrypt and decrypt PIN Session Keys. The PIN Master Key is generated outside the cryptographic boundary and input into the module. The PIN Master Key cannot be created through key agreement with the BlackBerry Enterprise Server™.
PIN Session Key	A Session Key that is specifically a Triple DES key used to encrypt and decrypt PIN-to-PIN data. The module generates PIN Session Keys using the implemented FIPS 186-2 RNG.
Software Integrity Key	An RSA public key used to verify the integrity of the module software.
EC Diffie-Hellman Key Pair	An elliptic curve key pair used to perform Diffie-Hellman key agreement.
EC MQV Key Pair	An elliptic curve key pair used to perform MQV key agreement.
HMAC SHA-1 Key	A key used to calculate and verify a keyed message authentication code using the HMAC SHA-1 algorithm.
HMAC SHA-256 Key	A key used to calculate and verify a keyed message authentication code using the HMAC SHA-256 algorithm.
HMAC SHA-512 Key	A key used to calculate and verify a keyed message authentication code using the HMAC SHA-512 algorithm.

Self-Tests

The following table describes the self-tests implemented by the module:

Table 4. Module Self-Tests

Test	Description
Software Integrity Test	The module implements an integrity test for the module software by verifying its 1024-bit RSA signature. The software integrity test passes if and only if the signature verifies successfully using the Software Integrity Key.
AES-256 CBC KAT	The module implements a known answer test (KAT) for AES-256 in the CBC mode of operation. The test passes if and only if the calculated output equals the expected output.
Triple DES CBC KAT	The module implements a KAT for Triple DES in the CBC mode of operation. The test passes if and only if the calculated output equals the expected output.
SHA-1 KAT	The module implements a KAT for SHA-1. The KAT passes if and only if the calculated output equals the expected output.
SHA-256 KAT	The module implements a KAT for SHA-256. The KAT passes if and only if the calculated output equals the expected output.
SHA-512 KAT	The module implements a KAT for SHA-512. The KAT passes if and only if the calculated output equals the expected output.
HMAC SHA-1 KAT	The module implements a KAT for HMAC SHA-1. The KAT passes if and only if the calculated output equals the expected output.
HMAC SHA-256 KAT	The module implements a KAT for HMAC SHA-256. The KAT passes if and only if the calculated output equals the expected output.
HMAC SHA-512 KAT	The module implements a KAT for HMAC SHA-512. The KAT passes if and only if the calculated output equals the expected output.
RSA Verify KAT	The module implements a KAT for RSA signature verification. The test passes if and only if the calculated output equals the expected output.
EC Diffie-Hellman KAT	The module implements a KAT for the Diffie-Hellman key agreement protocol using elliptic curve cryptography (ECC). Using known input, a shared secret is generated. The test passes if and only if the calculated shared secret equals the expected shared secret.
ECC Pair-Wise Consistency Test	The module implements a pair-wise consistency test for each newly created ECC key pair.
Continuous RNG Test	The module implements a continuous RNG test, as specified in FIPS 140-2, for the implemented RNG.
FIPS 186-2 RNG KAT	The module implements a KAT for the FIPS 186-2 RNG. The KAT passes if and only if the calculated output equals the expected output.

All self-tests except the ECC Pair-Wise Consistency Test and the Continuous RNG Test are executed during power-up without requiring operator input or action. The Software Integrity Test is the first self-test executed during power-up.

Invoking the Self-Tests

The operator may invoke the power-up self-tests by resetting the module as described in Roles, Services, and Authentication on page 5.

The operator may also invoke the all of the self-tests with the exception of the Software Integrity Test, ECC Pair-Wise Consistency Test, and Continuous RNG Test, by performing the following operations:

3. Click the **Options** icon in the Home screen. The **Options** list appears.
4. From the **Options** list, click the **Security** item. The security options screen appears.
5. Click the trackwheel. A menu appears.
6. Click the **Verify Security Software** menu item.

When the self-tests are executed in this manner, the module displays the list of self-tests that are being executed and their pass/fail status upon completion.

Mitigation of Other Attacks

The module is not designed to mitigate any specialised attacks, thus the FIPS 140-2 requirements for mitigation of other attacks are not applicable.

Glossary

AES	Advanced Encryption Standard
CBC	Cipher block chaining
CSP	Critical security parameter
DES	Data Encryption Standard
EC	Elliptic curve
FIPS	Federal Information Processing Standard
HMAC	Keyed-hashed message authentication code
IEEE	Institute of Electrical and Electronics Engineers
LCD	Liquid crystal display
LED	Light emitting diode
MQV	Menezes, Qu, Vanstone
PIN	Personal identification number
PKCS	Public Key Cryptography Standard
PUB	Publication
RIM	Research In Motion
RNG	Random number generator
RSA	Rivest, Shamir, Adleman
SHA	Secure Hash Algorithm
SHS	Secure Hash Standard
USB	Universal serial bus